CI/CD

CONTINUOUS INTEGRATION AND CONTINUOUS DELIVERY/CONTINUOUS DEPLOYMENT.

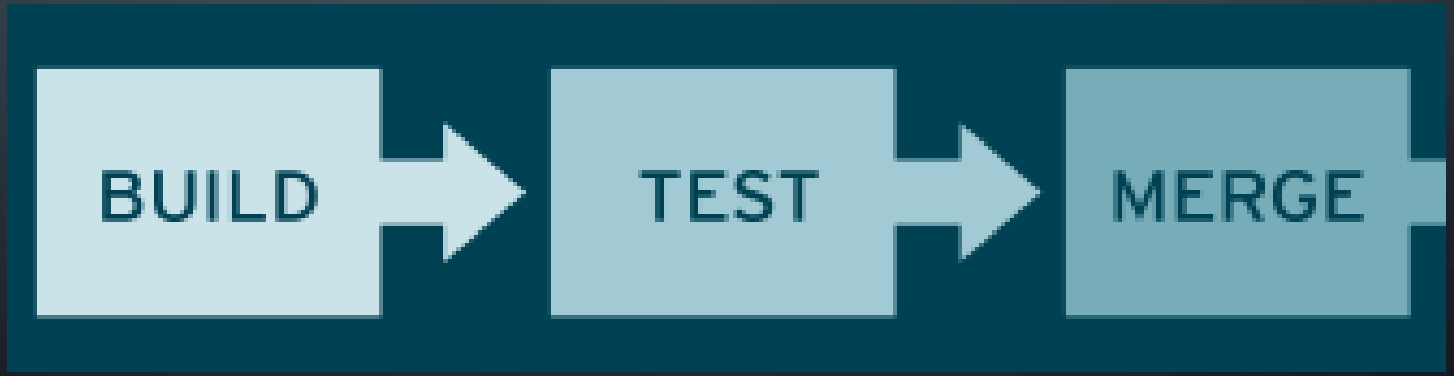
CONTINUOUS INTEGRATION

The practice of merging all developers' working copies to a shared mainline

several times a day to avoid conflicts in the code in the future.

It's the first step towards ensuring that we have a high quality, deployable artifact.

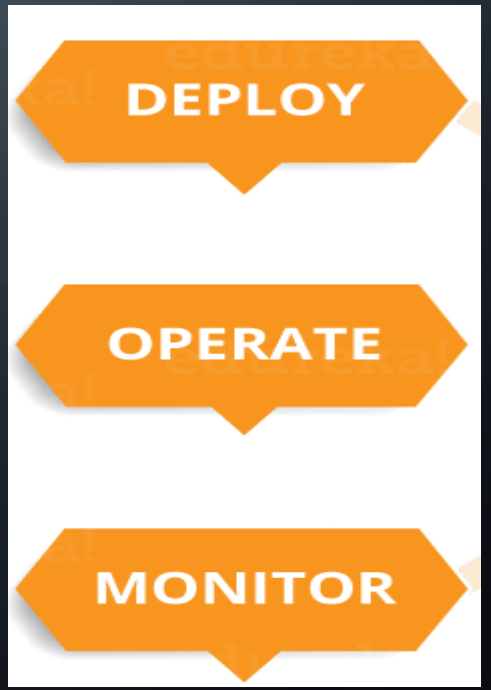
Some of the steps in this stage include compiling, testing, running static analysis, checking for vulnerabilities in our dependencies and storing the code artifacts.



CONTINUOUS DEPLOYMENT

This is the process by which verified changes in codebase or system architecture are deployed to production as soon as they are ready and without human input.

Some steps in this stage include setting up infrastructure, provisioning servers, copying files, smoke testing, promoting to production and even rolling back a change if something did not look right.



BENEFITS OF CI/CD TO OUR BUSINESS

1. Higher efficiency:

Increased productivity is one of the leading advantages of a CI/CD pipeline.

2. Reduced risk of defects:

Finding and resolving defects late in the development process is costly and time-consuming.

3. Faster product delivery:

With a smooth CI/CD workflow, multiple daily releases can become a reality.

4. Better planning:

Organizational designs must be adaptable to changing economic conditions.

BENEFITS OF CI/CD TO OUR BUSINESS (CONT.)

5. Log generation:

Observability is pivotal for DevOps. If something isn't right, you need to figure out why.

6. Quick rollback if required:

One of the most exclusive benefits of a CI/CD pipeline is that it leads to the quick and easy rollback of code changes if there are any issues in the production environment after a release.

7. Cost-effectiveness:

The CI/CD pipeline takes a different approach to software delivery.